EXHIBIT A

PowerCast Delivery

Program Manager	paulwa
Development	alicew
Test	pareshp
Product Planning	imranq
Product Design / Usability	v-kathym
Team	PowerPoint
Open Date	REDACTED
Current Status	Reviewed
Filename	powercast delivery.html

Contents

Summary

The goal of this feature is to enable the use of HTML as the data format for PowerCast conferencing scenarios. PowerCast is the code name for the PowerPoint-on-top-of-NetShow scenario. This will make it possible to use the browser as the client-side viewer so that users can participate on any platform. The presenter (who is the source of the content) will deliver the conference from PowerPoint, leveraging PowerPoint's presentation support tools. Here's an overview. Related specs exist for Delivery UI, Scheduling, and Web Pages. Look here for a glossary of PowerCast terms.

This feature consists of two different "parts": the first is the presenter's side of things, and the second is the audience's side of things. As mentioned above, the presenter will deliver the event from PowerPoint9. The audience will watch and listen to the event from their IE 4.0+ or Navigator 4.0+ browser. V3 browsers are not supported.

Design Goals and Justification

Using V4 HTML as the format for conferencing will provide the following advantages:

- Everyone, everywhere can view it. The audience can be on any platform and do not need to download a viewer.
- We would get a lot of technologies for free (e.g. streaming sound, streaming video and scheduling).
- It is easier for third parties to integrate with PowerPoint conferencing.

The PowerCast feature area is one of the key web collaboration features for PowerPoint9. This aspect of PowerCast heavily leverages Outlook, as well as the Outlook/NetMeeting work being done by Outlook. The PowerCast feature is built on top of NetShow.

The goal for this release is to create a complete, easy-to-use, and highly automated solution for intranet-based presentation. Going beyond the firewall presents too many problems to solve in this release. However, it will still be possible to deliver a presentation over the internet, but PowerPoint will not deliver the complete internet delivery solution.

The PowerCast solution in PP9 will obsolete the current NetShow Presenter addin. NetShow will stop development of the addin after NetShow 2.0.

Scenarios

1) A company meeting is broadcast to all desktops in the company. Audio, video, and slides are sent. Email is used to collect feedback and Q&A questions.

2) A sales training presentation is given to sales groups in remote sites. Each remote site is a conference room with 3-10 salespeople. Only slides are sent. A phone conference is used to narrate the presentation and to allow the sales teams to ask questions.

3) An Office status meeting is broadcast to the Office team. A series of presentations are sent, along with audio and video. Email and possibly Chat is used to collect feedback and questions.

4) BillG's Comdex keynote speech is broadcast over the internet. (In this case, Outlook is not used to schedule the event and setup is not fully automated.)

The presentation is scheduled in advance using PowerPoint (and Outlook). Before the presentation starts, I fire up PowerPoint on my computer, and start the broadcast with one step. From my computer I go through my presentation. When I'm done, the next presenter gives her slides from her computer.

The audience doesn't need to do anything special. They connect to the net from their desktops and laptops and view the presentation from their browser. They don't worry about having PowerPoint installed. They just kick back and watch the presentations.

Design

This document will cover the functionality required to enable the use of HTML content between PowerCasts driven by PowerPoint and viewed from a browser. Scheduling and UI are covered elsewhere: Delivery UI, Scheduling, and Web Pages.

The feature is based on HTML and browsers primarily to make it possible to be viewed from all major platforms (Win16, Win32, MacOS, UNIX). However, the source for the PowerCast event will be Win32-only since the source application is PowerPoint9.

Architecture

The PowerCast event will be delivered from PowerPoint9. This section covers specifics related to delivering the presentation

Overview

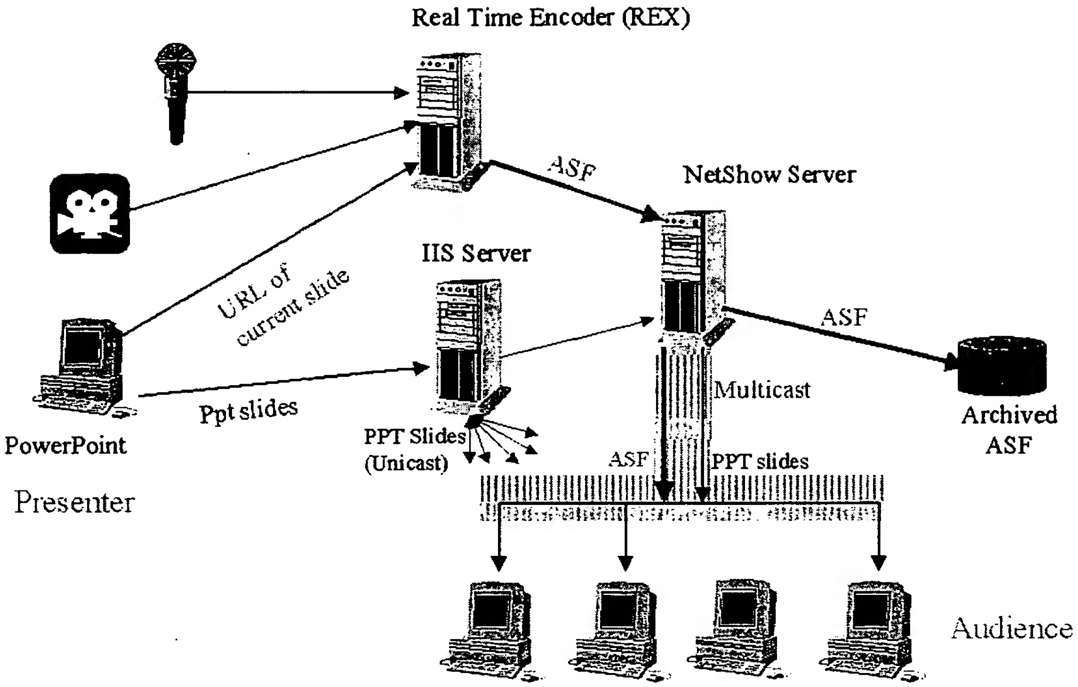


Figure 1: NetShow Overview

This diagram illustrates how the pieces fit together. PowerPoint will integrate the NetShow real-time encoder (REX). REX will be the only NetShow piece PowerPoint needs to directly interact with.

REX and PowerPoint can run on the same machine, Win95 or NT. The Web Server and the NetShow Server can run on the same machine, however NetShow Server requires NT Server. The NetShow Server will be a shared resource for everyone on the intranet.

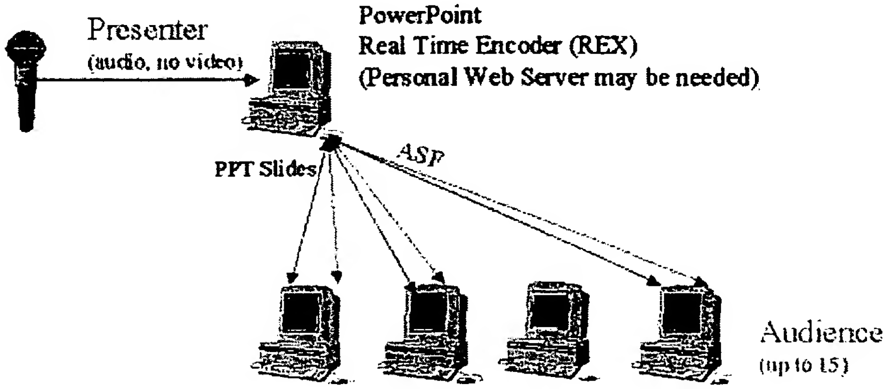


Figure 2: Small Conference scenario

PowerPoint 9 will make it possible to deliver small, audio only presentations without the need for a NetShow server. REX 3.0 will support up to 15 direct connections in an audio-only configuration.

Component List

The following is a list of components needed to enable delivery of the PowerCast.

Component	Where?	Installed by	Notes

PowerPoint 9	Presenter	Office	
Outlook 9	Presenter, Audience	Office	(optional)
REX 3.0 (w/ CODECS)	Presenter's machine or special A/V computer	Office	
NetShow Player 3.0 (w/CODECS)	Audience	Office/IE	
NetShow Server 3.0	NT Server	NetShow	(optional)
Web Server	Anywhere		(optional) IIS, or FrontPage Personal Server, etc.

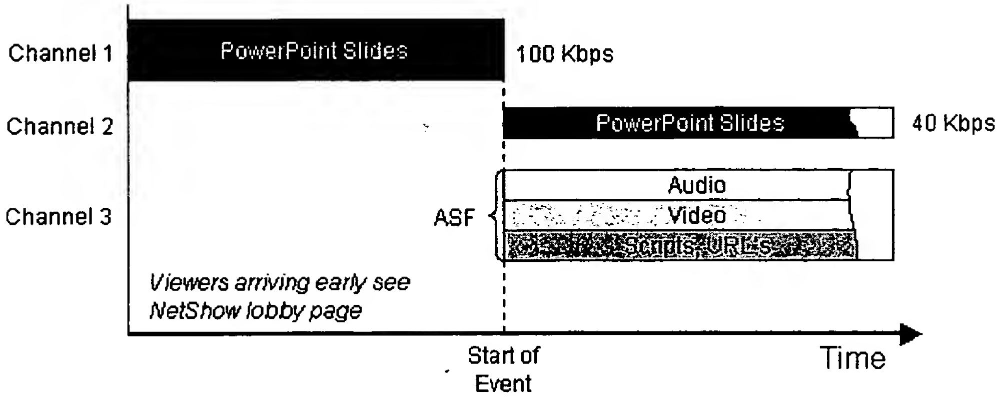


Figure 3: Multicast timeline, NetShow multicast of presentation

Before the event starts, the PowerPoint master slides are multicast out at a high transmission rate. Viewers who tune in early will see the NetShow lobby web page, while in the background, the browser cache will be getting pre-loaded with the PowerPoint slides by the NetShow File Transfer Control. At 100Kbps, 500KB worth of master slides will be pre-loaded in about 1 minute.

When the event starts, multicast of the ASF stream begins. Viewers are taken to the webcast page. The PowerPoint slides (not master slides) are multicasted at a reduced transmission rate. Viewers who tune in late must wait on the lobby page until they receive all of the master slide content.

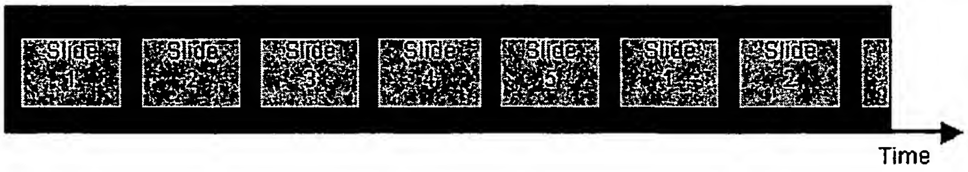


Figure 4: Multicast of PowerPoint slides

The transmission of the PowerPoint slides is independent of the ASF stream and independent of the current PowerPoint slide. What is being sent is actually all of the HTML files and supporting files (e.g. image files, sound files, etc.) that comprise the presentation. The files are multicast by the NetShow in alphabetical order, so to ensure that files are sent in presentation order, they should be named with leading 0's. The support files for each slide should be sent before the next slide. After all of the files have been transmitted, the server starts again at slide 1. This repeats infinitely until the presentation ends.

The NetShow 3.0 server does not have the ability to skip slides that have already been viewed. (Better file control is a candidate for NetShow 4.0).

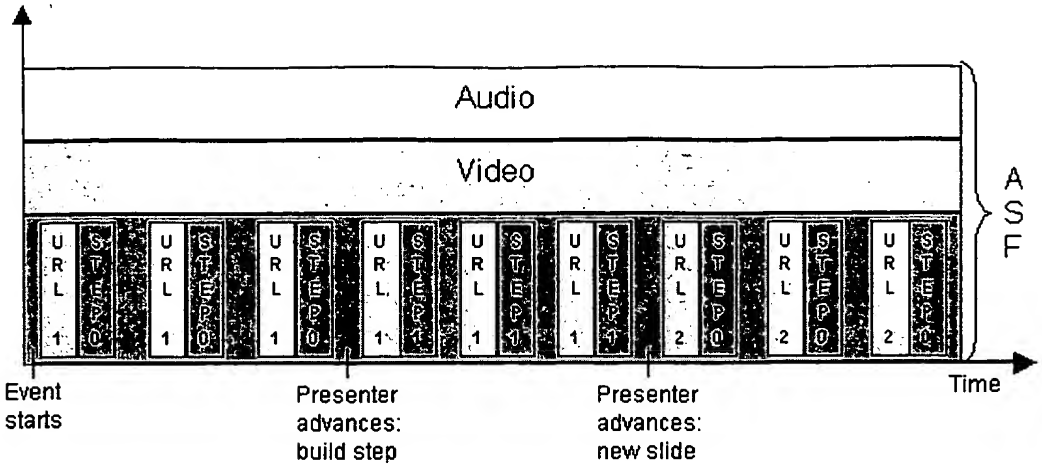


Figure 5: Multicast of ASF stream

The ASF stream is comprised of audio, video, and control strings. The three streams are combined into one (ASF) by the Real Time Encoder (REX). The three streams are always kept in sync. The ASF is decoded on the viewer's machine by the NetShow player. The player plays the audio and video, while sending the control strings off to the browser. New URL's are sent for a new slide change or build step. Each URL is followed immediately by the build step control string. These are used to animate the build steps in V4 browsers. The URL/control string pairs are repeated every 2 seconds to ensure multicast reception and to reduce sync-up time for late-joiners.

Behind the scenes

The following steps define what happens when the user schedules/starts a PowerCast.

1. ***User schedules the PowerCast using PowerPoint and Outlook***. Broadcast descriptions, options and file locations are determined and saved in the presentation and the broadcast.dat file. See Scheduling spec. Lobby page is created at this time. See Web Pages spec. NetShow controls, Lobby page, and broadcast.dat are all placed in the URL location.

2. ***Copy the lobby page to the NetShow server location.*** (*Only if using a NetShow server*). This verifies that we have write access to this location.

3. ***User selects Broadcast->Begin***. Either manually or Outlook does it automatically. The broadcast dialog is displayed.

4. ***PowerPoint saves HTML to file server location***. PowerCast HTML is saved. The file structure is flat. Settings from the Setup Show dialog are honored (e.g. slide range, custom show, animations), and settings from the broadcast options dialog are honored (e.g. show speaker notes).

5. ***Copy the HTML to the NetShow server location***. (*Only if using a NetShow server*).

6. ***Start the multicast programs***. (*Only if using a NetShow server*). At this point, viewers will only be listening to the ASF and the high-bandwidth file transfer. After the broadcast begins, viewers will be listening to the ASF and the low-bandwidth file transfer. The high-bandwidth file transfer multicasts out only the master slides, slide 1, 2, and 3. The low-bandwidth file transfer multicasts out only slides, not the master slides.

7. ***Start and stop REX to verify that it is properly installed and configured***.

8. ***Microphone check and/or camera check***. (*Only if using camera and/or mic*). See Delivery UI spec.

9. ***Wait for user to start the broadcast***.

10. ***Start REX***.

11. ***Start PowerPoint slideshow***.

12. ***Send URL's to REX***. Every two seconds, send a URL corresponding to the current slide and build to REX. This is sent in the ASF stream, keeping the audience in sync. A new URL is sent immediately after any slide change or build in PowerPoint.

13. ***Stop PowerPoint slideshow***.

14. ***Stop REX***.

15. ***Stop the multicast programs***. (*Only if using a NetShow server*).

16. ***Copy html files to the archive location***. (Only if user specified this option).

17. ***Delete files from NetShow server location***. (Only if using a NetShow server). Don't do this if it is the same location as the archive location.

18. ***Delete files from the file server location***. Don't do this if it is the same location as the archive location. Only delete the PowerPoint slides. Don't delete the lobby page, etc.

PowerCast HTML

This is covered in detail in the Broadcast HTML spec. We will use a special Broadcast version of HTML output so that editing data will be stripped from the output, reducing file size, and special scripts for the broadcast are included. We will save only V4 HTML format. The prefixing scheme allows us to multicast only master slides and slides 1-3 before the broadcast starts (?1*.*), and then to multicast slides but not masters after the broadcast begins (S*.*).

The current HTML naming plan is as follows:

File Type	File Name	Part of PowerCast
Master Slides	M1_nn.htm	Yes
Slides 1 - 3	S1_nnnn.htm	Yes
Slides 4 - n	SX_nnnn.htm	Yes
Graphics	<prefix>_imagennnn.jpg	Yes
Sounds	<prefix>_sndnnnn.wav	Yes
Start page	Default.htm	Yes
Roundtrip edit data	Editdata.mso	No
Non-viewing edit data	Pres.xml	??

Custom shows and slide ranges are supported. The slides that are not part of the custom show or slide range should not be saved in the broadcast html, otherwise viewers would be able to manually get to them and possibly uncover private information.

Notes pages should only be saved if that option is selected in Broadcast options. Notes text is not visible in the webcast pages, but is visible when the user selects "View previous slides".

All animation effects will be converted to "appear" effects. That's because fly effects have serious performance problems and most other animation effects are not supported.

In the REX-only case, we will use a trick, such as <IMG src="foo.htm"> in order to pre-load slides on the audience machines. This should not be used when broadcasting with a NetShow server.

Interaction with REX

Need to know: 1) Location of REX, 2) .ASD settings, 3) URL's to send. These will be specified as part of the scheduling process.

The registry will contain a default machine name for a remote REX machine. This is only used in the case that REX is not running on the local (presenter's) machine. The user can override the default when scheduling the PowerCast. This setting is saved in the document when scheduling the event. The new machine name will be overwrite the default in the registry unless the admin has it locked.

When the broadcast begins, PowerPoint needs to start a new REX. REX may be running locally or on a remote machine. We won't connect to an already running REX, like the NetShow Presenter allows you to do. This would add too much complexity (however it would allow you to broadcast a series of presentations). When starting REX, we need to give it the correct initialization settings. These are contained in an .ASD file.

We pass the location of the .ASD file to REX when starting REX.

The .ASD file contains encoder specific information to control Rex. NetShow 3.0 will provide us with two preset .ASD files called System Stream Formats. One will be audio-only for 110K networks, one will be audio + video for 110K networks. We will choose the appropriate .ASD file based on the audio and video settings in the Broadcast options dialog.

REX timing is based on audio input, so a sound card is always required for REX to run. This does not mean that a microphone is required. And in order to disable sound, we need to mute the input.

Issue: We need to modify the .asd file to include the archive settings.

Issue: Should we add reg entries for which ASD files to use?

After starting slideshow, PowerPoint will begin sending URL's and control scripts indicating the current slide and the current build number. We will need to record the base URL, i.e. the web or file server location where we put the PPT slides. The individual slide names are appended to the base URL.

The URL/control string pairs should be re-sent every 2 seconds. This will minimize the amount of time that viewers will have to wait when the tune in late. This will also help in cases when the first URL is lost due to reliability problems. Loss of packets from the ASF stream will not noticeably affect the audio or video.

The URL that is sent out should be a combination of URL + build step so that late joiners can sync up. The build step should be absolute, not relative. The URL's that we use will be relative URL's, not absolute. This allows the html to be moved for on-demand playback of the broadcast.

The automation API's for REX 3.0 will be the same as for REX 2.0.

DCOM needs to be present when REX is running on Win95. It is installed with OSR2. We will need to install it in earlier versions of Win95.

When slideshow ends, REX is stopped. The broadcast ends.

Information that we will use to start REX:

Field	Default Value
[Description]	
Use Script	0x00000001
Use Video	0x00000001 Broadcast Options
Video Codec	0x3467706D
Image Width	0x000000B0
Image Height	0x00000090
Image Quality	0x00000051
Color FOURCC	0x32595559
Color Depth	0x00000010
Frame Rate	0x0000000F
Seconds per IFrame	0x00000005
Bitrate	0x0001B800
ECC Span	0x0000000A
Audio Concealment	0x000007D0
Use Audio	0x00000001 Broadcast Options
Audio Codec	0x00000055
Audio Format	02Km000000000sCG0s0200Qm1206a0T00l07C0B00W0340CG0i0300CW0r0200I01w02m0801D06y0RW
Description	02BW000000001eJG1G04K0Hm0W04m0OG1v06K0SW0W0380801X07K0P01f06y03G0A04q0K01504S0D0

[Encoder Configuration]	
Use Dedicated Port	0x00000001
Use Stored Source	0x00000000
IP Port	0x00001B5F Registry
MCM Server Name	020W0000000002000
Stream Alias	020W0000000002000
Audio Source	02A0000000000wLW1f06G0PG1r06q0801N0640TW1b0200KW1b06C0Rm1o06G0801Y07a0801N06a0RW
Video Source	026G000000000gLW1f06G0PG1r06q0801M06a0P01b06y080130640S01q07K0SW1b0000
Input FileName	020W0000000002000
Stream Window	0x00000005
Enable Network	0x00000001
Save Locally	Broadcast\Archive: False Broadcast Options
ASF FileName	Broadcast\ArchiveDir: Null Broadcast Options
ASF File Length	Broadcast\MaxArchiveSize: 50000 (KB)
ASF File Duration	0x00000000

Issue: Performance is very poor, even on a P200, when PPT Slideshow is running while REX is compressing video without hardware assistance. We need to either require hardware assisted video compression or allow for REX to run on a separate machine. We can also have REX automatically scale the video quality based on the horsepower of the machine. Different codecs require different amounts of CPU power. We can also specify a low video rate and size, "talking head" video codec.

The NetShow team is providing us with an audio and a video codec.

REX 3.0 does not duplicate script commands, like 2.0 did. This is the desired behavior since PPT repeats the script every 2 seconds.

Interaction with File Server or Web Server

The HTML files need to be saved to a location where they can be accessed by all viewers of the PowerCast. This will typically be a file server, but it may also be a web server In theory, the html could be hosted in a shared directory on the presenter machine, but that would require a lot of CPU horsepower. Additionally, archived files would not available for on-demand viewing if the presenter machine is offline.

Audience machines will receive the html via a direct file connection or possibly a http connectionIt would be up to the user to create the shared directory.

The default location will be stored as a reg setting. It should be a file/web server that everyone has access to. An admin will set this default for all users in a particular group. This setting can be updated at any time by the admin. The registry value will be of the form \\alto\user\broadcasts. We will append to that the user name and a unique folder name, resulting in something like \\alto\user\broadcasts\paulwa\broadcast23456. The string "broadcast" is a resource, so we can localize it. The unique id is based on the current date, e.g. 980228. If that folder already exists, we will append to it 980228-1, 980228-2, 980228-3, ...980228-25, until an available one is found. Adding the username creates some organization within the default folder. Adding the "broadcast" string identifies the contents of the subfolders, rather than just having a bunch of random numbers.

The user can override the default location when scheduling and setting up the PowerCast. This location is saved in the document when scheduling the event. The new location will be overwrite the default in the registry unless the admin has it locked.

This directory will be created when the event is scheduled, and the lobby page will be saved there at the same time.

When the broadcast dialog is invoked, the broadcast html is saved to this location. If any files existed in that location before, they should be deleted before saving again. However, we shouldn't delete the lobby page files. Delete only SX*.*, S1*.*, and M1*.*.

Interaction with NetShow Server

If a NetShow server is being used, three things need to be done:

1) The HTML files need to be copied to a location where they can be accessed by the File Transfer Program running on the NetShow server.

2) The File Transfer Programs need to be started on the NetShow server.

3) The ASF Multicast Programs need to be started on the NetShow server.

PPT needs to know the file location for the HTML files, and the machine name of the NetShow server. Defaults for these settings will be provided in the registry by a network admin. The user can override the defaults when scheduling the PowerCast. These settings are saved in the document when scheduling. The new settings will overwrite the defaults in the registry unless the admin has them locked

Files for the NetShow server are handled just like files for the file server. When the broadcast dialog is invoked, files are copied from the file server location to the NetShow file location. This may be the same location, in which case the copying does not need to be done. Old files should be deleted before copying new ones there. PowerPoint should only copy the files that are to be multicast, i.e. don't copy *.mso or pres.xml to the NetShow file location. The client File Transfer Control, that runs in the browser and receives the multicast of files, will be set to check file modification dates so that older cached files are replaced by new ones.

The NetShow server will need to be configured by the admin to give access rights to presenters in the group so that they can start programs on the server. This needs to be noted in the ORK.

To start the File Transfer and ASF programs on the NetShow server, the server itself must already be running. There are three programs to start: high bandwidth file transfer, low bandwidth file transfer, and the ASF transfer. These programs need to know: the source for the HTML files; the source for the ASF stream (REX location); the destination URL. The high bandwidth transfer program will use ?1*.* to broadcast master slides and slides 1 thru 3. The low bandwidth transfer program will use S*.* to transfer slides, but not master slides.

A proxy dll, NSLite, is used by PowerPoint to remotely launch these multicast programs on the NetShow server. One part of NSLite runs on the PowerPoint machine and the other part runs on the NetShow server. NSLite may need to be updated with NetShow 4.0, but it will maintain backwards compatibility with PowerPoint. NSLite has been developed and will be maintained by the NetShow team.

The programs are started when the Broadcast dialog is invoked. Canceling the Broadcast dialog stops the programs. Ending the slideshow also stops the programs.

The proxy interface includes specifying a "kill time" for the programs. This ensures that the programs will not be left running in the case that the presenter machine crashes or the network connection is lost.

The NetShow Server may be configured to handle Unicast rollover. That means that if users cannot access the multicast (perhaps because their network is not multicast enabled), they can still receive the presentation broadcast via unicast. This will be stored as a reg setting set by the admin.

Values used to start File Transfer multicast:

Property	Value	Description
Name (!)	Broadcast\NetShow\Options\SessionName: Null	The name of the session. The default is null. If specified, it must be unique among the sessions on the server, as well as globally in order to be able to recreate the session later. If not, a globally unique name is generated.
Title (!)	Schedule dialog	The human-readable title of the session. The default is the same as the Name property.
Description	Schedule dialog	The textual description of the session. The default is null.
Author	Presenter	The author's name. The default is null.

Copyright	Broadcast\NetShow\Options\Copyright: Null	The copyright notice of the content. The default is null.
Unicast Rollover	Broadcast\NetShow\UnicastRollover: False	Specifies whether to perform unicast rollover or not. The default is no.
Source Base URL (*)	Broadcast\NetShow\NetShowFileLoc: None Server Options (NetShow settings)	The base URL or UNC where the slides are. Wildcards characters are allowed to defines the source file names. The default is null (invalid). Note that the files in the subdirectory, if any, will not be transferred.
Output Base URL (*)	Broadcast\BrowserLoc: None Server Options (File server settings) (Same as ASF Base URL)	The base URL that the client will recognize as when the files are finally transferred to the client machine. The source file names are used to complete the URL by concatenating with the base URL. The default is null. This property is used to pre-load the client's URL cache with these files. Note that this property must be set if you want the files to go into the URL cache on the client's machine. If this property is set, the Output Base Directory property will be ignored.
Output Base Directory (* if no Output Base URL)	N/A	The base directory in the client where the files will be transferred to. The default is %TEMP%, which means the files will go into the temporary directory of the client defined by the TEMP environment parameter. Note that this parameter is ignored if the Output Base URL property is set.
Redundancy Ratio	Broadcast\NetShow\ FECRedundancyRatio: 10 (%)	The percentage of how much data redundancy to be transferred. Using the unreliable transfer protocol, sending redundant data increase the probability that the client would get the data completely. In the intranet, where packet loses are minimal, this can be small. The default is 20%.
Data Bandwidth	Broadcast\NetShow\ FileTransferHiBandwidth: 110 (kbps) Broadcast\NetShow\ FileTransferLoBandwidth: 40 (kbps)	The maximum data transfer rate. This is specified in Kbps. The default is 256.
Contact Address	Broadcast\NetShow\Options\ContactAddress: Null	The session's contact address. The default is null.
Contact Phone Number	Broadcast\NetShow\Options\ContactPhone: Null	The session's contact phone number. The default is null.
Contact Email	Broadcast\NetShow\Options\ContactEmail: Null Schedule Dialog	The session's contact email. The default is null.
Multicast Address	Broadcast\NetShow\Options\MulticastAddress: Null	The IP multicast address used for broadcasting. The default is null. If specified, it must be a valid multicast IP address, unique among other addresses used on the server. If not, an address will be generated.
Multicast Port	Broadcast\NetShow\Options\MulticastPort: Null	The port used for broadcasting. The default is null. If specified, it must be a valid port, unique on the IP address

		used on the server. If not, a port number will be generated.
Multicast TTL	Broadcast\NetShow \MulticastTTL: 5	The multicast time-to-live. The default is 1 (for Intranet). This is the number of 'hops' the multicast packets can make before reaching the destination.
Drop-Dead Time	Broadcast\NetShow\Options\DropDeadTime: Null (hours)	The date and time when the session should already be done. If the session has not been deleted by then, the system will delete it. The default is null (24 hours after it is created). This property makes sure that the server can clean up if for some reason the user didn't.

Values used to start ASF multicast:

Property	Value	Description
Name (!)	Broadcast\NetShow\Options\SessionName: Null	The name of the session. The default is null. If specified, it must be unique among the sessions on the server, as well as globally in order to be able to recreate the session later. If not, a globally unique name is generated.
Title (!)	Schedule dialog	The human-readable title of the session. The default is the same as Name property. Note that the name will appear as title on the client's player.
Description	Schedule dialog	The textual description of the session. The default is null. Note that the description will appear on the client's player.
Author	Presenter	The author's name. The default is null.
Copyright	Broadcast\NetShow\Options\Copyright: Null	The copyright notice of the content. The default is null.
Rex Address (*)	Broadcast\REXComputerName: Null (use this machine) Broadcast Options	The IP address or the name of the machine where Rex is running. The default is null. This property must be set if you want the server to connect to Rex directly. If this property is set, the Rex Alias will be ignored.
Rex Port	Broadcast\REXPort: 7007	The port on the machine to use to communicate with Rex. The default is 7007
Rex Alias (* if no Rex Address)	N/A	The alias that is used to find the Rex address. The default is null. This property must be set if you want the server to connect to Rex via the alias. If the Rex Address is set, this property will be ignored.
ASD UNC	Broadcast\NetShow\Options\ASDUNC: Null	The URL where the ASD file is used to configure Rex. The default is null, which means the stream format is one of the SSF.
Unicast Rollover	Broadcast\NetShow\UnicastRollover: False	Specifies whether to perform unicast rollover or not. The default is no. Note that the unicast manager is assumed to be installed on the same machine as the NetShow services.
Base Directory (*) (!)	Broadcast\FileServerLoc: Null	Directory path name, in UNC or local file format, where

	Server Options (File server settings)	the system can generate and store files that must be accessed by the clients. The NSC and the ASX files required by the session will be created here. The default is null (invalid).
Base URL (*) (!)	Broadcast\BrowserLoc: None Server Options (File server settings) (Same as FTS Output Base URL)	Base URL for the client to access, equivalent of the Base Directory property. The client will access the NSC and ASX files from this base URL. The default is null (invalid).
Client Log URL	Broadcast\NetShow\Options\ReadWriteAdminURL: Null	URL that client can use to generate log of its activities and statuses. The actual logging is implemented by a cgi script behind this URL. The default is null (no log created).
Contact Address	Broadcast\NetShow\Options\ContactAddress: Null	The session's contact address. The default is null.
Contact Phone Number	NetShow\Options\ContactPhone: Null	The session's contact phone number. The default is null.
Contact Email	Broadcast\NetShow\Options\ContactEmail: Null Schedule dialog	The session's contact email. The default is null.
Auto Archive	Broadcast\Archive: False Broadcast Options	Specifies whether the content should be automatically archived.
Auto Archive Directory	Broadcast\ArchiveDir: Null Broadcast Options	Directory path name where the archive file is generated. Valid only when the Auto Archive property is set. The default is null, which is invalid when the Auto Archive property is set.
Auto Archive Size	Broadcast\MaxArchiveSize: 50000 (KB)	The file size limit of the archive file. The default is 0 (unlimited).
Multicast Address	Broadcast\NetShow\Options\MulticastAddress: Null	The IP multicast address used for broadcasting. The default is null. If specified, it must be a valid multicast IP address, unique among other addresses used on the server. If not, an address will be generated.
Multicast Port	Broadcast\NetShow\Options\MulticastPort: Null	The port used for broadcasting. The default is null. If specified, it must be a valid port, unique on the IP address used on the server. If not, a port number will be generated.
Multicast TTL	Broadcast\NetShow \MulticastTTL: 5	The multicast time-to-live. The default is 1 (for Intranet). This is the number of 'hops' the multicast packets can make before reaching the destination.
Drop-Dead Time	Broadcast\NetShow\Options\DropDeadTime: Null (hours)	The session should be done this many hours after it was created. If the session has not been deleted by then, the system will delete it. The default is null (24 hours after it is created). This property makes sure that the server can clean up if for some reason the user didn't.

Small Conference (REX-only) Scenario

This is our out-of-the-box solution for Office users who do not have a NetShow server. It is also useful for smaller, ad hoc meetings where video is not important, but simplicity is.

REX and PowerPoint will both run on the same machine (NT or Win95). If audio is being used, then the microphone will be connected to this machine as well. No other machines are required to deliver the presentation, however a file server is strongly recommended.

There is no multicast of files or ASF in this configuration. REX 3.0 will support up to 15 direct connections for sending ASF. The REX port number will be written into the global.js file and used by the netshow player to connect directly to REX.

Since there is no multicast of files, there will be no pre-caching of the html on the audience machines. In order to minimize the synchronization delay for the audience, we will always pre-load the next slide. However, since IE5 will not support pre-load, we will need to use a trick, such as <IMG src="foo.htm">. This will download the htm file, and if it is put into a 0x0 image tag off the screen, you can put it into the cache and it won't display. Works like preload. This tag should only be used in the REX-only scenario. It should not be used when broadcasting with a NetShow server.

Note that there's not really a way to determine how many people are invited or to restrict access. You may send email to an alias inviting hundreds of people. There's no way to know how many intend to attend the meeting in person and how many via NetShow. And how many will be viewing on the same machine, e.g. in a conference room. Connections to REX are on a first-come first-served basis. The 16th person to tune in will be denied access.

Our minimal machine for this configuration: 16M P60 Win95, or 32M P60 NT

Our target machine: 32M P133 Win95, or 32M P133 NT

Issue: Need to see what kind of performance is necessary to host all of this on one machine. That may reduce the max number of connections from 15. We could also scale this number based on CPU power, but that would increase the testing matrix. NetShow is investigating.

Interaction with Client (NetShow Player)

The browser will be launched from the Outlook reminder. Launching the browser a few minutes before the presentation starts gives the multicast time to pre-cache the presentation. Other than that, this will work just as it does now with NetShow Player 2.0. The NetShow player will be automatically downloaded when the viewer goes to the NetShow event web page.

The NetShow client and the PowerCast web pages do not require active server support or web server support.

How does the NetShow Player handle out of disk space conditions or other cases where the browser cache is full? Brian Crites: *Our File Transfer Service that multicasts files to the client use the APIs in the INetSDK to put the files into the cache. It is the logic of the cache that is responsible for deciding when the files in the cache are stale and when files need to be deleted due to reaching the limits imposed by the options set by the user. As files are pushed to the client, prior files are automatically removed as necessary by the browser.*

A typical presentation will be about 3MB, which will have no problem fitting in the browser cache.

Beta 2.0 versions of the Win16, Mac and 4 Unix versions are available now on http://www.microsoft.com/netshow/download/betaplayers.htm

Final 2.0 versions of the Win16 and Mac players should be available by REDACTED and final Unix players by REDACTED.

All the codecs that are used by the System Stream Formats will be cross-platform.

The URL is exposed in the meeting invitation. If the browser is closed during the presentation, the user can get back to the broadcast page by using the browser history, or by opening up the calendar in Outlook, or by retyping the URL. If users in a conference room want to tune in, they will need to manually enter the URL (since that machine would not receive an invitation).

The 3.0 player for Win32 will be ready on REDACTED. Beta versions for Mac and Win16 will also be ready on REDACTED. Final dates for Mac, Win16 and Unix are not yet known.

Note that the NS Player is only in-place on Win32.

Archiving for On-Demand

As the event is being distributed live, it can also be saved so that it can be viewed after the fact. This scenario is referred to as "on-demand" in NetShow's syntax so it'll be used here.

The ASF will be saved to the specified archive location. The default location for archiving broadcasts will be specified by an admin and stored in the registry. This includes a max file size. The registry also includes a setting for a life-span for the archive files, but it is not currently used. The user can override the default location in the broadcast options dialog.

The ASF contains relative URL's pointing the html files, which means that the html files and the ASF file can be moved, e.g. to a CD, without editing the ASF.

Issue: How do we gracefully handle out-of-disk-space issues? Compare against what PP97 currently does for Voice Narration.

Issue: Do we need a visible option for max size? Maybe just some text. Can we convert the disk size to minutes for the user?

Netshow 3.0 Features

Netshow 3.0 is scheduled for REDACTED. New features include the following

Server:

Security -- Can restrict access to a NetShow.

REX:

Ability to preview video

Preset .ASD files (system stream formats)

No longer need to copy .asd files to server

Auto-select video, pixel format, etc.

Same automation API's as 2.0

Minimum install (no SDK's). This will be about 1.5M.

Feature Team Links

PowerCast

NetShow - Jim Durkin (PUM), AmirM (GPM), MikeBeck (GPM 3.0, 4.0), RamiroC (PM Client), BretOr (PM Server), Richard Saunders (PM Tools), LauraLa (PM Web samples), Steven Levy (Dev Mgr), Brian Crites (Dev Lead), J. Burnett (Test Mgr)

Netshow Home Page http://microsoft/netshow/

Further information on Multicast technology: http://msr/groups/BARC/telepresence/

Worldwide, Localization, Far East

See http://officeweb/Specs/powerpoint/PowerCast Delivery/powercast file list.htm

User Assistance and Discoverability

Explain key user scenarios, user terms, and Answer Wizard index entries (e.g. include WordPerfect or 1-2-3 terms). Give examples of queries and the topics they should return. This section should also include the methods by which the user will discover this feature.

Programmability

Compatibility and Platform Requirements

PowerCast requires for presenting:

- PPT9
- Win95, Win98, NT4, or NT5.
- REX 3.0
- Minimal machine for presenting w/o NetShow server: 16M P60 Win95, or 32M P60 NT machine
- Target machine for presenting w/o NetShow server: 32M P133 Win95, or 32M P133 NT machine
- Network connection
- Sound card (REX requires this, even if you are not sending audio).

Optional for presenting:

- NetShow Server 3.0
- A Web Server
- Microphone
- Video camera
- Video compression h/w
- Outlook
- Internet connection

PowerCast requires for viewing:

- NetShow Client 3.0
- A V4 browser, such as IE4 or Nav 4
- Win 9x, NT, Mac, or Unix
- Intranet connection

Optional for viewing

- Sound card
- Outlook

NetShow will be backward compatible with previous releases, and the old releases will ignore ASF content they don't understand. Verified REDACTED by EricFl.

Setup and Administration

Files to install:

http://officeweb/Specs/powerpoint/PowerCast Delivery/powercast file list.htm

Reg Settings:

All of these reg settings will be pushed to clients via policy. That allows NetShow to be installed/configured/updated after Office has been installed.

http://officeweb/Specs/powerpoint/PowerCast Delivery/powercast reg entries.htm

Q&A

These might be "rude" Q&A or they might be questions that the average person might have after reading this specification. Please include answers to dumb questions too.

Issue List

See issues identified in context above.

From PaulPo:

Can we have a rehearse mode so the presenter can try out the whole scheme to ensure things work well, e.g. am I speaking too soft, too loud, etc. This can be done by doing a dry run and reviewing the archive.

Can we provide some kind of feedback to the presenter as to how many people are currently tuned in? This allows the presenter to delay the presentation if no one is tuned in, or close down the presentation quickly if everyone has left.

From Ralf:

My ongoing concern is that we deliver on true one-step usability for the presenter. If we don't this will end up being only used for Meidenbauer type events (i.e. 1% of the shows in a corporation.)

As a presenter, assuming I have a designated server for my workgroup, I should be able to schedule and deliver a PowerPoint+audio (one way, no audience feedback) show from my desk without the intervention of any technicians. Scheduling a presentation in Outlook should send invites, create the liason web page, etc. At delivery time, I should just go to Outlook, select the meeting, and say "deliver this." Everything should get set up for me, PPT should boot with the right file, and I just start flipping slides and talking into my PC's microphone. No questions asked, no A/V technician required.

Even the MIS setup should be optional. If MIS specified a server for my workgroup to use, great. If not, the code should make smart choices for me as the presenter. If the machine I am using when scheduling the meeting is an NT machine with the NetShow server installed Netshow should use that. If the machine is a win95 machine or an NT machine with no server the code should let me browse the net for servers (the list should hopefully be refined to servers that I have the right to use; I presume there is such a distinction.) And of course the server I pick should become the default for the next time I present.

If we do our jobs right the presenter should have to try real hard for a broadcast to fail, as opposed to having the presenting process be an obstacle course where any one of a number of steps can go wrong and block the whole show. (This may sound hard core. But we should remember that people are failing at delivering shows with our built-in conferencing, which is immensely simpler than NetShow. We should err on the side of making delivering presentations an absolutely brainless exercise.)

Last but not least, we should thoroughly usability both the presenter and the viewer side of this.

Cut Items

Placing a NetShow server on microsoft.com for "demo" purposes. Hooking up to ISP's to deliver a NetShow presentation. Both of these are cut due to the complexities introduced by going through firewalls.

[Issue: JPEG images could be written using restart markers to tolerate the loss of packets during the transfer of the image to the client machines. This eliminat6s the need for re-transfer from the server via http. NetShow would provide PowerPoint with code to generate loss-tolerant JPEG images.] This is not important now that we are sending the slides in HTML.

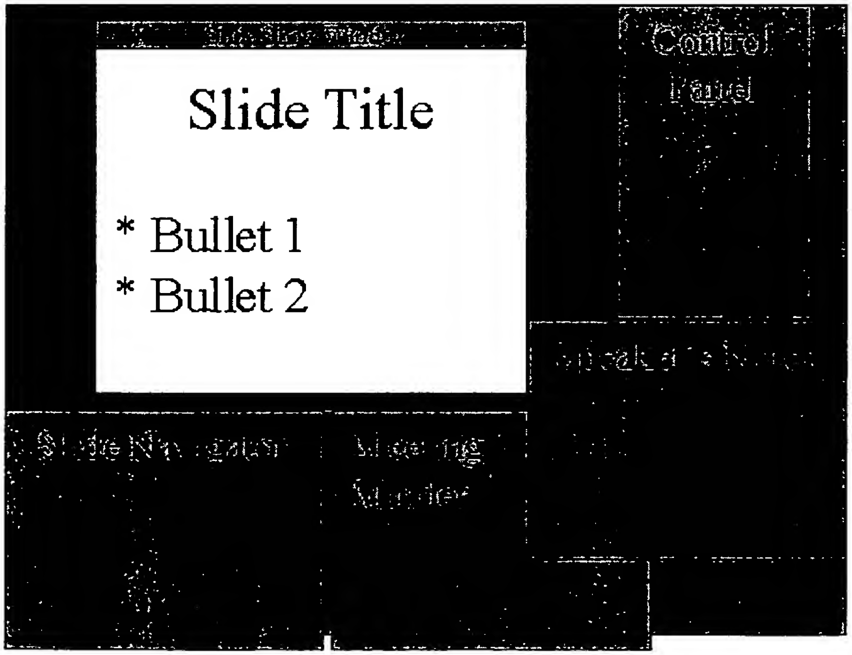
Issue: Will we allow the presenter to black out the screens of the viewers, as in PP97?

Issue: How can the files be removed automatically, including the HTML files on the web server? Maybe this is a utility that NetShow should provide. (NetShow).

This page should not be necessary since we will be re-transmitting the current URL every 30 seconds.

☒ REDACTED FIGURE ENTITLED -RELIABILITY_PLEASE_WAIT-

View on Two Screens scenario:



Change History

Date	Changes Made	Program Manager
REDACTED	Created	JohnT and ImranQ
REDACTED	Added details	JohnT
REDACTED	Added diagram for presenter-side of things.	JohnT
REDACTED	Updated some issues.	JohnT
REDACTED	Added on-demand section	JohnT

REDACTED	Last updates prior to handing off to PaulWa.	JohnT
REDACTED	Updates with new architecture	PaulWa
REDACTED	Started adding presenter UI	PaulWa
REDACTED	Incorporated feedback from first meeting	PaulWa
REDACTED	Created new spec for UI	PaulWa
REDACTED	Review meeting feedback	PaulWa
REDACTED	Meeting w/Rich Saunders. Fixed links.	PaulWa
REDACTED	Lots o' updates	PaulWa
REDACTED	Updates based on the meeting with Audi & Rich Saunders, plus the NetShow Lite interface. Also added lots of reg entries.	PaulWa
REDACTED	Added spreadsheet with registry entries and file lists. Added sections on broadcast html and behind the scenes steps. Added details to REX, NetShow, file server, client interaction sections.	PaulWa
REDACTED	Delete html files when done with broadcast. All animation effects will be converted to "appear" effects. Pre-load trick should not be used when broadcasting with a NetShow server. Added .ASD file contents. Updated File Transfer table and ASF table.	PaulWa
REDACTED	Only delete html for ppt slides, not everything else. REX requires a sound card. We will use relative URL's. REX 3.0 does not duplicate script commands (this is good). Base Directory is a required field for starting the ASF program. Win32 NS Player 3.0 will be ready on REDACTED. Betas for other platforms ready on REDACTED.	PaulWa